



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/814,563	03/31/2004	Zhibin Wang	ORACL-01513US0	6910
80548	7590	07/06/2009		
Fliesler Meyer LLP 650 California Street 14th Floor San Francisco, CA 94108			EXAMINER MITCHELL, JASON D	
			ART UNIT	PAPER NUMBER
			2193	
			MAIL DATE	DELIVERY MODE
			07/06/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/814,563	Applicant(s) WANG ET AL.	
	Examiner JASON MITCHELL	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 April 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-4,6-11,13-15,18 and 20-26 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-4,6-11,13-15,18 and 20-26 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

This action is in response to a request for continued examination filed on 4/8/09.

Claims 1-4, 6-11, 13-15, 18 and 20-26 are pending in this application.

Response to Arguments

Applicant's arguments filed 4/8/09 have been fully considered but they are not persuasive.

In the last full par. on pg. 10 the applicants state:

... Claim 1, as amended, defines that each software test tool is operable to test a plurality of different graphical user interfaces for a plurality of different software applications. Applicant respectfully submits that the plurality of test tools disclosed in McNeely are each device-specific, and are not operable to test different devices. As such, Applicant respectfully submits that McNeely, in view of Dubovsky, does not disclose or render obvious that each software test tool is operable to test a plurality of different graphical user interfaces for a plurality of different software applications, as defined by Claim 1.

The examiner respectfully disagrees. As noted in the rejection McNeely's abstraction of device specific test languages (see e.g. col. 3, lines 53-67 "generalized test environment") can easily be modified to provide an abstraction of tool specific but application independent test tool languages such as those described by Dubovsky (par. [0048] "scriptable GUI test tool 10"). To do so the only aspect of McNeely's system which would change is the specific details of the translation (e.g. McNeely col. 15, lines 47-52 "based on the mapping provided by the appropriate communication interface package, interprets the command within the context of the specific [test tool] to which

Art Unit: 2193

the command refers”). Doing so would produce a system with the same benefits as the original McNeely system.

In the first par. on pg. 12 the applicants state:

Claims 7, 14, and 21 define wherein any of the software test tools can be removed and replaced with another software test tool. It was asserted in the Office Action that McNeely discloses this feature. Applicant respectfully traverses this assertion. McNeely discloses that communication with GUI-based devices can occur via a graphical user interface if a suitable GUI tester is added via a new package. (Column 13, lines 47-49). McNeely appears to describe a test tool that can test multiple devices on a network. Since each device may have its own device-specific testing language, McNeely suggests using a tool that is device-generic. McNeely further appears to disclose that the test tool can also test GUI-based devices by adding a new package. Accordingly, Applicant respectfully submits that McNeely discloses device-specific software test tools and a single device-generic tool for testing multiple devices. The software test tools described by McNeely do not appear capable of testing different devices; and therefore cannot be replaced with another software test tool since the result would be nonfunctional. As such, Applicant respectfully submits that McNeely, in view of Dubovsky, does not disclose or render obvious that any of the software test tools can be removed and replaced with another software test tool, as defined by Claims 7, 14, and 21.

The examiner respectfully disagrees. McNeely's disclosure at col. 13, lines 47-49 (“a suitable GUI tester is added via a new package”) indicates that a ‘new package’ is all that is required to support the addition of a test tool. Because these packages provided the necessary data to perform the translation to the device or, in the case of the asserted combination the test tool, specific language (see e.g. McNeely col. 15, lines 47-52 “based on the mapping provided by the appropriate communication interface package, interprets the command”) when it is no longer necessary or desirable to test a particular device or use a test tool the associated package can be removed. This results

Art Unit: 2193

in no loss of functionality that is not also found in the claim (i.e. by removing and replacing a test tool the removed test tool can no longer be used).

In the 2nd par. on pg. 12 the applicants state:

Claims 22-24 define that the system defines a contract interface for use as an entry point in loading the libraries corresponding to the plurality of different software test tools, and wherein additional software test tools that use a different scripting language can be dynamically plugged into the system at the entry point by defining an execution interface of those additional software test tools to comply with the contract interface. In the Office Action, it was asserted that McNeely discloses this feature. Applicant respectfully traverses this assertion. McNeely appears to disclose packages that include procedures to enable a user to create a test case using generic commands. The procedures in the package appear to access the device specific commands. However, McNeely does not appear to disclose either a contract interface for use as an entry point or an execution interface of the additional software test tools that complies with the contract interface. As such, Applicant respectfully submits that McNeely, in view of Dubovsky, does not appear to disclose or render obvious these features.

The examiner respectfully disagrees. First it is noted that no explicit definition of a contract interface is found in the specification nor do applicants arguments express a specific perceived distinction between the claimed contract interface and the cited passage in McNeely (e.g. col. 11, lines 1-33). The examiner understands the claimed 'contract interface' to represent a collection of functionalities which any compliant library must implement. This understanding is supported, for example, by disclosures in US 6,449,618 to Blott et al. (e.g. col. 13, lines 52-58) and US 2002/0178438 to Arbousov et al. (e.g. par. [0061]).

Based on this understanding McNeely's disclosure (col. 11, lines 1-33) appears to describe the claimed 'contract interface' (i.e. requires "proc startup { }" and "proc cleanup { }" and those of ordinary skill will understand that such procedure definitions

Art Unit: 2193

require defined 'entry points' at which to begin execution of the procedure (e.g. "The procedures access the device specific packages"). Further it should be recognized that by identifying these entry points to the generic test system the system can begin to make use of the 'plugged in' library. Accordingly McNeely's disclosure anticipates a reasonably broad interpretation of the claim.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-3, 6-10, 13-15 and 20-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 7,117,411 to McNeely et al. (McNeely) in view of US 2003/0055836 to Dubovsky (Dubovsky).

Regarding Claims 1, 8 and 15: McNeely discloses a system that provides a generic user interface testing framework, comprising:

a computer including a computer readable medium, and a processor operating thereon (Fig. 1);

a plurality of different software test tools, wherein each software test tool is associated with a different tool-specific scripting language (col. 13, lines 57-62 "device-specific commands ... may be tool command language commands"), that can be

Art Unit: 2193

invoked by a user to perform testing operations (col. 13, lines 49-52 “a plurality of device-specific test case packages 404”; col. 13, lines 47-49 “a suitable GUI tester is added via a new package”), and wherein each of the plurality of different software test tools use only their own tool-specific scripting language (col. 15, lines 54-60 “tool command language command (ST6)”) to test the plurality of different devices (col. 13, lines 47-49 “communication with GUI-based devices can occur”);

a test case input file stored on the computer readable medium, that contains a plurality of generic interface commands that are abstractions independent of any of the tool-specific scripting languages (col. 15, lines 47-52 “an abstract command language command (ST4)”), wherein the test case input file can be edited and reused as necessary by the user to specify different generic interface commands for testing in any of the different software test tools (col. 4, lines 30-34 “test case and test plan editor”); and

an interpretive engine that executes on the computer, and that includes a plurality of dynamically loaded libraries corresponding to the plurality of different software test tools (col. 13, lines 49-52 “a plurality of device-specific test case packages 404”), and including at least one library for each of the plurality of different software test tools, wherein each library is a group of functions written in each tool-specific scripting language (col. 15, lines 36-40 “the appropriate communication interface packages associated with each DUT”), wherein the interpretive engine receives the generic interface commands defined in the test case input file, identifies which libraries are required, loads the required libraries associated with the software test tool the user is

Art Unit: 2193

currently using, maps the generic interface commands to the software test tool's associated tool-specific scripting language (col. 15, lines 47-52 "based on the mapping provided by the appropriate communication interface package, interprets the command within the context of the specific DUT to which the command refers"; this requires identification and loading of the interface packages / libraries), uses the software test tool to perform the testing operations on the software application's graphical user interface using the associated tool-specific scripting language (col. 15, lines 54-60 "produce an equivalent tool command language command"; col. 13, lines 47-49 "Communication with GUI-based devices can occur ... if a suitable GUI tester is added via a new package"), and reports to the user the success or failure of the testing operations (col. 3, lines 53-56 "executing ... test cases"; col. 16, lines 6-8 "the resulting tool command language command is subsequently passed to the communication interface 420").

McNeely does not disclose a software application source code, including a graphical user interface as part of the software application or abstracting a plurality of application independent test tools.

Dubovsky teaches testing a plurality of different graphical user interfaces for a plurality of different software applications (par. [0015] "test case generation, maintenance and execution required during the development and test cycle of a GUI software project"; par. [0048] "A single test engine script 18 ... for each GUI application to be tested")

Art Unit: 2193

using a test tool and corresponding tool specific scripting language (par. [0048] “the scripting language of the scriptable GUI test tool 10”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply McNeely’s “generalized test environment” (see e.g. col. 3, lines 53-67) to testing a plurality of different graphical user interfaces for a plurality of different software applications as taught by Dubovsky (see e.g. par. [0015] & [0048]) by substituting Dubovsky’s application / GUI independent test tools (par. [0048] “the scriptable GUI test tool 10”; par. [0007] “There are several known testing tools for debugging GUI applications”) for McNeely’s device specific test tools (col. 13, lines 49-52 “a plurality of device-specific test case packages 404). Those of ordinary skill in the art would have been motivated to do so in order to save developer time and resources (McNeely col. 3, lines 53-67 “the operator need only be familiar with a common script language rather than device-specific test commands”; Dubovsky par. [0016] “reduce the investment in manpower to implement, maintain and enhance automated test software”) by providing a generic test scripting environment for such systems (McNeely col. 3, lines 53-67; Dubovsky par. [0007] “There are several known testing tools for debugging GUI applications”).

Regarding Claims 2 and 9: The rejections of claims 1 and 8 are incorporated respectively; further McNeely does not explicitly disclose the software test tools stored locally on the same computer or machine.

McNeely's background teaches that "The client/server framework allows a client to be located on any system in the network, even on the same system on which the server resides" (col. 3, lines 7-10).

Accordingly, it would have been obvious to one of ordinary skill in the art at the time the invention was made to store the software test tools on the same computer or machine as McNeely's "Test Tools Server" (see Fig. 3).

Regarding Claims 3 and 10: The rejections of claims 1 and 8 are incorporated respectively; further McNeely discloses the software test tools are stored at another computer or machine (Fig. 3).

Regarding Claims 6, 13 and 20: The rejections of claims 1, 8 and 15 are incorporated respectively; further McNeely discloses the test case input file is created offline and subsequently communicated to the interpretive engine (col. 15, lines 31-34 "downloads the test to execution engine 400").

Regarding Claims 7, 14 and 21: The rejections of claims 1, 8 and 15 are incorporated respectively; further McNeely discloses a software test tool can be replaced with another test software tool (col. 13, lines 47-49 "a suitable GUI tester is added via a new package"), but does not explicitly disclose the test software tool can be removed.

McNeely teaches “the test cases are independent of the number or types of devices under test” (col. 3, lines 56-57).

Accordingly, it would have been obvious to one of ordinary skill in the art at the time the invention was made to remove test software tools which had been replaced with new test software tools (col. 13, lines 47-49 “a suitable GUI tester is added”).

Regarding Claims 22-24: The rejections of claims 1, 8 and 15 are incorporated respectively; further McNeely discloses, wherein the system defines a contract interface for use as an entry point in loading the libraries corresponding to the plurality of different software test tools (col. 11, line 1-33 “The procedures access the device specific packages for multiple devices being tested and perform the functions for each specific device For example, the “proc startup{ }” ... proc cleanup{ }”), and wherein additional software test tools that use a different scripting language can be dynamically plugged into the system at the entry point by defining an execution interface of those additional software test tools to comply with the contract interface (col. 13, lines 47-49 “a suitable GUI tester is added”).

Regarding Claim 25: The rejection of claim 1 is incorporated; further McNeely discloses each software test tool is used only for execution of the test case input file,

Art Unit: 2193

and the test case input file is built independently of any software test tool (Test management system client 214 ... allows a user to ... construct[] and edit[] test plans”).

Regarding Claim 26: The rejection of claim 1 is incorporated; further McNeely and Dubovsky do not explicitly teach a first tool-specific scripting language associated with a first software test tool is mapped to a second tool-specific scripting language associated with a second software test tool, enabling test cases written in the second tool-specific scripting language to be executed by the first software test tool. However such a mapping is clearly achievable, at least by using McNeely's abstract command language as an intermediate form (col. 15, lines 47-52 “an abstract command language command (ST4)”). Further the use of a first tool-specific scripting language in place of McNeely's 'abstract command language' would not change the basic functionality of McNeely's generic tool (col. 15, lines 47-52 “based on the mapping provided by the appropriate communication interface package, interprets the command within the context of the specific DUT to which the command refers”). Specifically, all that would change would be the data represented in the mapping (col. 15, lines 47-52 “the mapping provided by the appropriate communication interface package”). Accordingly this does not represent a patentable distinction over the cited prior art.

Claims 4, 11 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 7,117,411 to McNeely et al. (McNeely) in view of US 2003/0055836 to Dubovsky (Dubovsky) in view of US 6,823,522 to Lamb (Lamb).

Regarding Claims 4, 11 and 18: The rejections of claims 1, 8 and 15 are incorporated respectively; further McNeely discloses a module for mapping the testing operations to generic interface commands (col. 15, lines 47-52 “based on the mapping provided by the appropriate communication interface package, interprets the command within the context of the specific DUT to which the command refers”).

The McNeely-Dubovsky combination does not explicitly disclose a rules-based wizard guiding the user to edit or create the test input case.

Lamb teaches a rules-based wizard that guides the user to edit or create the test case input file by choosing the testing operations to be included in the test case input file (col. 7, lines 13-16 “the developer is guided through the build process with assistance of a wizard which provides available options for each step of the build process”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made include Lamb's wizard (col. 7, lines 13-16 “a wizard which provides available options for each step of the build process”) in McNeely’s editor (col. 4, lines 30-34 “test case and test plan editor”). Those of ordinary skill in the art would have been motivated to do so in order to facilitate development of the test cases (McNeely col. 12, 2-5 “Editor 314 ... providing users with an easy to use and intuitive file creation/modification

Art Unit: 2193

environment”; Lamb col. 6, line 64-col. 7, line 7 “guide a developer through the application generation process”).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JASON MITCHELL whose telephone number is (571)272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason Mitchell/
Primary Examiner, Art Unit 2193

Application/Control Number: 10/814,563
Art Unit: 2193

Page 14